Factoranalyse met meerdere factoren

Psychometrie

Inhoudsopgave

1 Factoranalyse met meerdere factoren

1.1 Inleiding

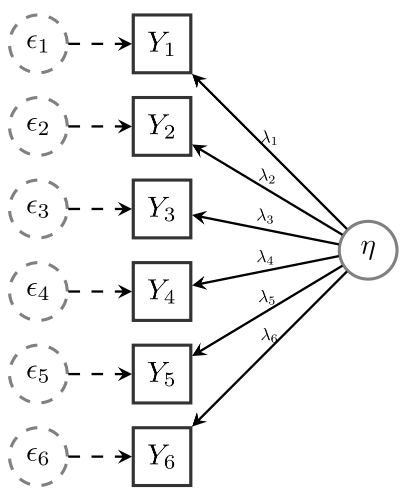
- de vier combinaties van de moderne 'latente variabele' benadering

	η numeriek	η categorisch
y numeriek	**factoranalyse**	latente profiel modellen
y categorisch	latente trek modellen (IRT)	latente klasse modellen

- bij factoranalyse beschouwen we zowel de latente variabelen ($\eta_1, \eta_2, \ldots$) als de indicatoren als numerieke metrische variabelen
- factoranalyse is met voorsprong de meest gebruike vorm van de latente variabele benadering

1.2 Voorbeelden van factormodellen

1-factor congeneriek model

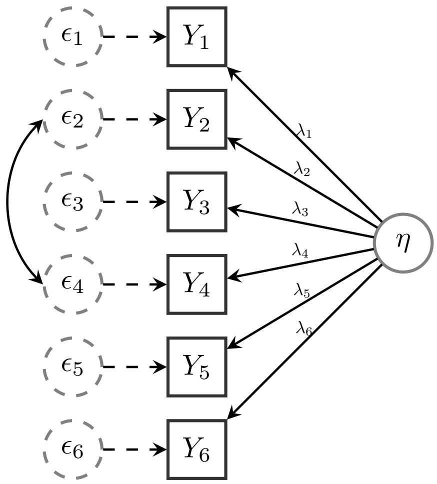


- dit model postuleert 1 factor, als enige oorzaak van de indicatoren
- behalve de (unieke) errorterm, wordt elk item enkel beïnvloed door die ene factor (de items zijn congeneriek)
- de factorladingen kunnen we samenvatten in de zogenaamde 'Lambda' matrix:

$$\Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix}$$

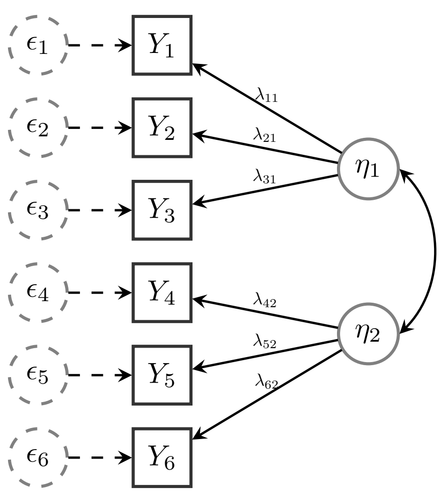
- de rijen zijn de indicatoren, de kolommen (hier slechts 1 kolom) de factoren

1-factor (niet-congeneriek) model



- items Y_2 en Y_4 hebben –naast de gemeenschappelijke factor– nog iets anders met elkaar gemeen, maar we weten niet wat
- in het kader van testconstructie is dit een belangrijk signaal: er is nog iets meer aan de hand met deze twee items
- mogelijks moeten (inhoudelijke) experten de items opnieuw evalueren, en desnoods nieuwe bedenken
- alternatieve verklaring: mogelijks volstaat 1 factor niet

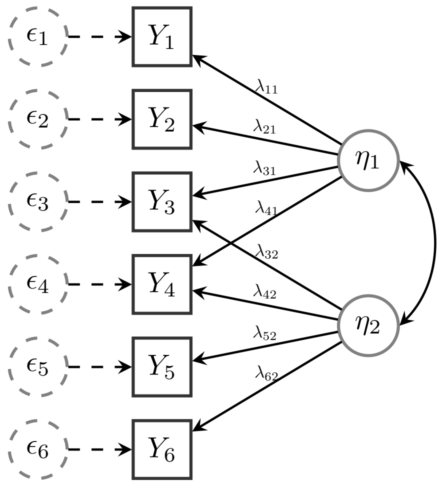
factormodel met twee gecorreleerde factoren



- dit is een factormodel met twee gecorreleerde factoren
- de eerste drie items beschouwen we als indicatoren van de eerste factor; de laatste drie items als indicatoren van de tweede factor
- dit model is ‘factorieel eenduidig’ (Engels: ‘factorial simple’): elke indicator hangt samen met slechts 1 factor; gerelateerd aan het concept ‘simple structure’
- de ‘Lambda’ matrix bevat nu twee kolommen:

$$\Lambda = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ 0 & \lambda_{42} \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \end{bmatrix}$$

factormodel met twee gecorreleerde factoren + kruisladingen

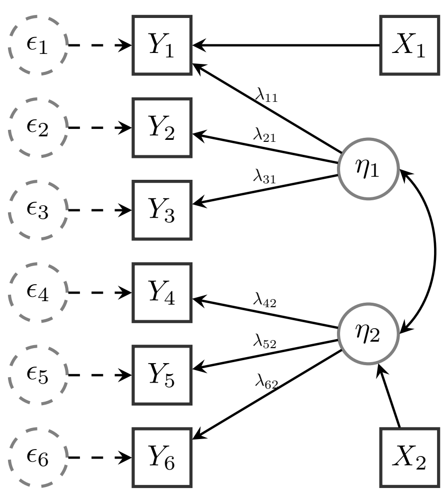


- dit model bevat twee ‘kruisladingen’: Y_3 en Y_4 zijn indicatoren van zowel η_1 als η_2
- bij testconstructie is dit een signaal dat de twee items ambigu zijn

- wellicht moeten (inhoudelijke) experten de items opnieuw evalueren, en desnoods nieuwe bedenken
- dit model is ‘factorieel complex’: er is geen eenduidige relatie tussen de indicatoren en de factoren
- de ‘Lambda’ matrix:

$$\Lambda = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \end{bmatrix}$$

mimic model

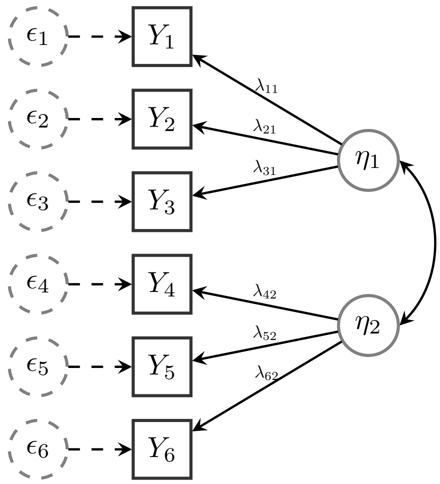


- een ‘mimic’ model (multiple indicators, multiple causes)

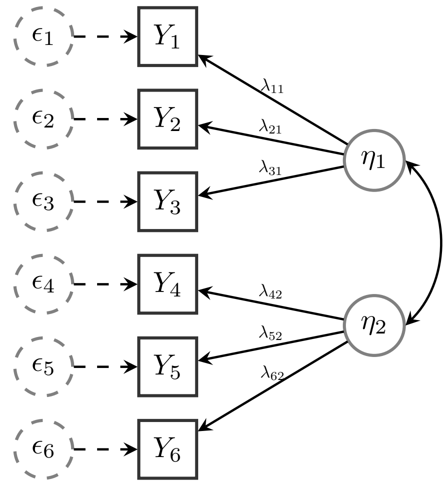
- X_1 en X_2 zijn exogene geobserveerde variabelen (bv. geslacht, leeftijd)
- indien een rechtstreeks (significant) effect op een indicator: indicator is niet ‘equivalent’ voor de verschillende waardes van X (item bias)
- indien een rechtstreeks effect op een factor: mogelijke verklaring voor de variabiliteit van de factor
- eigenlijk een voorbeeld van een structureel vergelijkingsmodel (SEM)

‘multiple group’ factormodel

Groep 1

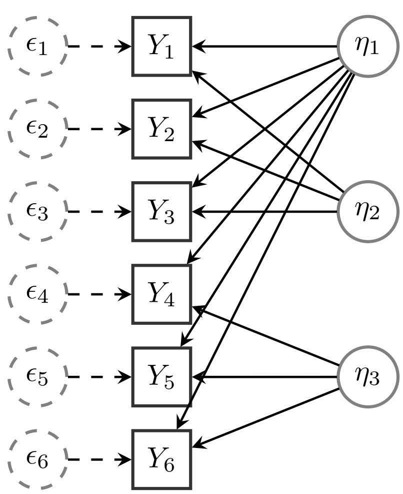


Groep 2



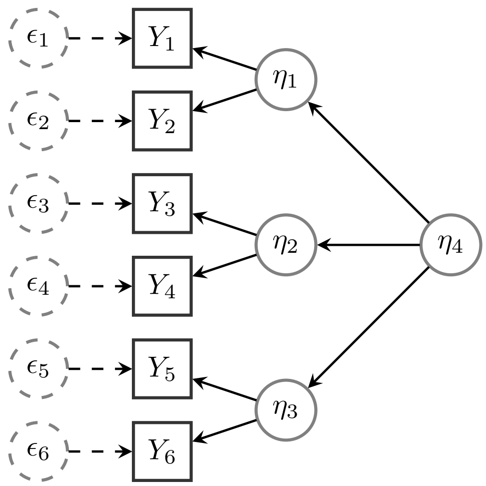
- in hoeverre zijn de modelparameters (factorladingen, intercepten, correlaties, ...) dezelfde in beide groepen?
- het probleem van meetinvariantie: is de relatie tussen de indicatoren en de factoren dezelfde in beide groepen?

bifactor model



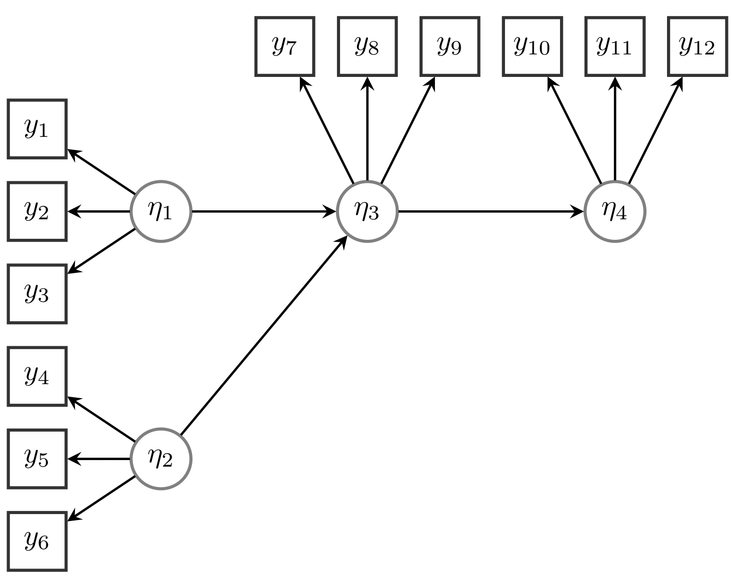
- een ‘globale’ factor (η_1) heeft een invloed op alle indicatoren
- twee specifieke factoren (η_2 en η_3) hebben enkel een invloed op een subset van indicatoren
- de factoren zijn *niet* gecorreleerd (‘orthogonaal’)
- de specifieke factoren (η_2 en η_3) corresponderen niet altijd met psychologische constructen, maar met methode-gerelateerde effecten (‘common-method variance’) omdat de scores van sommige items via dezelfde methode werden verzameld (bv. zelfrapportage)
- hoewel conceptueel heel aantrekkelijk, zijn bifactor modellen heel controversieel

tweede-orde factormodel



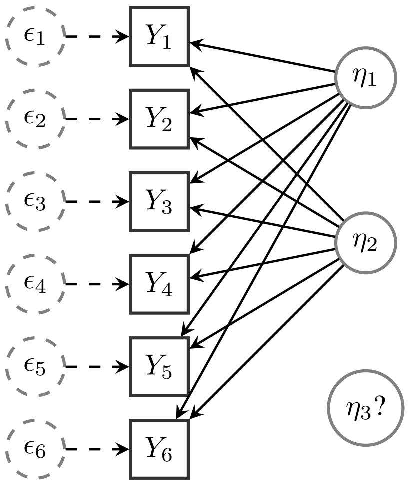
- η_1, η_2 en η_3 zijn eerste-orde factoren
- η_4 is een tweede-orde factor
- tweede-orde (en hogere orde) factormodellen zijn heel populair in vele deelgebieden van de psychologie (persoonlijkheidspsychologie, intelligentie, ...)
- in de intelligentieliteratuur correspondeert η_4 vaak met de g-factor, en is (bijvoorbeeld) η_1 = verbale intelligentie, η_2 = visuo-spatiale intelligentie, en η_3 = geheugencapaciteit
- (optioneel: de Lambda matrix bevat enkel de eerste-orde factoren en heeft dus drie kolommen; de tweede-orde factoren –en hun factorladingen– komen in een andere matrix terecht)

structureel vergelijkingsmodel



- een structureel vergelijkingsmodel (Engels: 'Structural Equation Model', SEM) postuleert (mogelijks causale) effecten tussen latente variabelen
- wellicht het meest relevante statistisch model voor de sociale- en gedragswetenschappen

exploratief factormodel



- aantal factoren ligt nog niet vast
- we willen weten welke items 'samenhoren' ('laden' op dezelfde factor)
- in tegenstelling tot confirmatorische factormodellen (waar vele factorladingen op voorhand op nul worden gefixeerd) worden alle factorladingen geschat
- rotatie is een essentieel onderdeel van exploratieve factoranalyse
- Lambda matrix (indien 2 factoren, en zonder identificatie-restricties)

$$\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \\ \lambda_{51} & \lambda_{52} \\ \lambda_{61} & \lambda_{62} \end{bmatrix}$$

1.3 Exploratief versus confirmatorisch

- exploratieve factoranalyse (EFA):
  - hoeveel factoren zijn er nodig?
  - welke items hangen samen met welke factor? en dus waarvoor staan de factoren? wat is hun interpretatie?
  - courant in software (ook SPSS), en heel populair
- confirmatorische factoranalyse (CFA):
  - gegeven een factormodel (met een vast aantal factoren, en hun relaties met de indicatoren), past het model bij de data?
  - was tot voor kort minder courant in software (niet in SPSS), en daarom minder gebruikt
  - veel meer mogelijkheden (bifactor, mimic, sem), en doorgaans te verkiezen boven exploratieve factoranalyse
- in het speciaal geval van 1 factor is EFA = CFA

1.4 Exploratieve factoranalyse

de ingrediënten van een EFA

- we beschikken over P scores, afkomstig van een (grote) steekproef met N observaties
- EFA vertrekt gewoonlijk vanuit de correlatiematrix van de P scores
- uitvoering EFA:
  1. selecteer een methode om de factoren te extraheren ('factorextractie')
  2. bepaal hoeveel factoren er moeten worden weerhouden
  3. indien meer dan 1 factor: roteer de factoroplossing teneinde een betere interpretatie van de factoren te bekomen

(optioneel) factor extractie

- maximum likelihood
  - assumptie: indicatoren zijn multivariaat normaal verdeeld
  - bonus: goodness-of-fit indices, standaardfouten (en confidentie-intervallen) voor de factorladingen (enkel met SEM software)
  - te verkiezen indien de EFA wordt opgevolgd door een CFA
  - nadeel: leidt soms tot een 'improper solution' (bvb. geen convergentie, of communaliteiten hoger dan 1.0); doch dit is meestal een signaal dat er ernstige problemen zijn met het factor model of de data zelf
- principal factors (of 'principal axis factoring')
  - vrij van assumpties omtrent de distributie van de indicatoren (robust)
  - geen 'goodness-of-fit' indices
- bemerk: principale componentenanalyse (PCA) is geen 'factor extractie' methode (niettemin de default in SPSS)

voorbeeld: 8 subschalen omtrent persoonlijkheid

- N=250 personen die psychotherapie volgen (outpatients) werden gescoord op 8 subschalen uit een persoonlijkheidstest (Bron: Brown (2006), figuur 4.1)

  1. anxiety (n1)
  2. hostility (n2)
  3. depression (n3)
  4. self-consciousness (n4)
  5. warmth (e1)
  6. gregariousness (e2)
  7. assertiveness (e3)
  8. positive emotions (e4)

- men verwacht twee factoren: neuroticisme en extraversie
- de data is reeds samengevat in een correlatiematrix

invoeren van de correlatiematrix

```
> N <- 250
> Brown2006 <- matrix(c(
+    1.000,  0.767,  0.731,  0.778, -0.351, -0.316, -0.296, -0.282,
+    0.767,  1.000,  0.709,  0.738, -0.302, -0.280, -0.289, -0.254,
+    0.731,  0.709,  1.000,  0.762, -0.356, -0.300, -0.297, -0.292,
+    0.778,  0.738,  0.762,  1.000, -0.318, -0.267, -0.296, -0.245,
+   -0.351, -0.302, -0.356, -0.318,  1.000,  0.675,  0.634,  0.534,
+   -0.316, -0.280, -0.300, -0.267,  0.675,  1.000,  0.651,  0.593,
+   -0.296, -0.289, -0.297, -0.296,  0.634,  0.651,  1.000,  0.566,
+   -0.282, -0.254, -0.292, -0.245,  0.534,  0.593,  0.566,  1.000),
+   nrow=8, ncol=8,
+   dimnames=list(c("n1","n2","n3","n4","e1","e2","e3","e4"),
+                 c("n1","n2","n3","n4","e1","e2","e3","e4")))
> Brown2006

       n1     n2     n3     n4     e1     e2     e3     e4
n1  1.000  0.767  0.731  0.778 -0.351 -0.316 -0.296 -0.282
n2  0.767  1.000  0.709  0.738 -0.302 -0.280 -0.289 -0.254
n3  0.731  0.709  1.000  0.762 -0.356 -0.300 -0.297 -0.292
n4  0.778  0.738  0.762  1.000 -0.318 -0.267 -0.296 -0.245
e1 -0.351 -0.302 -0.356 -0.318  1.000  0.675  0.634  0.534
e2 -0.316 -0.280 -0.300 -0.267  0.675  1.000  0.651  0.593
e3 -0.296 -0.289 -0.297 -0.296  0.634  0.651  1.000  0.566
e4 -0.282 -0.254 -0.292 -0.245  0.534  0.593  0.566  1.000
```

Model 1: 1 factor – factor extractie: maximum likelihood

```
Call:
factanal(factors = 1, covmat = Brown2006, n.obs = N)

Uniquenesses:
   n1    n2    n3    n4    e1    e2    e3    e4
0.225 0.290 0.289 0.240 0.807 0.841 0.841 0.867

Loadings:
   Factor1
n1  0.880
n2  0.842
n3  0.843
n4  0.872
e1 -0.439
e2 -0.398
e3 -0.399
e4 -0.364

               Factor1
SS loadings      3.599
Proportion Var   0.450

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 367.57 on 20 degrees of freedom.
The p-value is 1.06e-65
```


interpretatie

- alle items liggen op 1 'as': de neuroticisme items liggen aan de positieve kant, de extraversie items liggen aan de negatieve kant
- deze oplossing is niet geroteerd, maar 1-factor oplossingen kunnen we niet roteren
- de neuroticisme items domineren (relatief hoge factorladingen, lage unieke errorvarianties)
- de communaliteit is het stuk van de variantie (van een geobserveerd item) dat verklaard wordt door de factor
- de communaliteiten voor de 8 items zijn hier gelijk aan het kwadraat van de factorladingen:

```
> as.vector(fit$loadings)^2

[1] 0.7750551 0.7096992 0.7108506 0.7603457 0.1926589 0.1586661 0.1593542
[8] 0.1328093
```

```
> round(sum(fit$loadings^2), 3)

[1] 3.599

> round(sum(fit$loadings^2) / 8, 3)

[1] 0.45
```

- de `SS loadings 3.599` betekent: de som van de gekwadrateerde factorladingen (sum of squares: SS) (of communaliteiten) is gelijk aan 3.599
- de `Proportion Var 0.450` betekent: 45% van de totale variantie van de 8 items wordt verklaard door de factor (de som van de 8 communaliteiten gedeeld door 8)
- de modeltoets $\chi^2(20) = 367.57, p \approx 0$ is (sterk) significant: we kunnen de nulhypothese ('het model past goed bij de data') *niet* behouden

Model 2a: 2 factoren, zonder rotatie

```
Call:
factanal(factors = 2, covmat = Brown2006, n.obs = N, rotation = "none")

Uniquenesses:
   n1    n2    n3    n4    e1    e2    e3    e4
0.219 0.280 0.289 0.217 0.362 0.293 0.381 0.511

Loadings:
   Factor1 Factor2
n1  0.850   0.240
n2  0.809   0.254
n3  0.816   0.213
n4  0.839   0.282
e1 -0.568   0.562
e2 -0.543   0.642
e3 -0.530   0.582
e4 -0.475   0.513

               Factor1 Factor2
SS loadings      3.872   1.577
Proportion Var   0.484   0.197
Cumulative Var   0.484   0.681

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 9.58 on 13 degrees of freedom.
The p-value is 0.728
```

interpretatie

- dit is (voorlopig) een niet-geroteerde oplossing
- de extraversie items laden zowel op de eerste als de tweede factor
- de modeltoets $\chi^2(13) = 9.58, p = 0.728$ is niet langer significant, wat suggereert dat een twee-factor oplossing een goede beschrijving is van de data
- op de lijn `Proportion Var` lezen we af: de eerste factor verklaart 48.4% van de totale variantie; de tweede factor verklaart 19.7% van de totale variantie
- op de lijn `Cumulative Var` lezen we af: de twee factoren verklaren samen 68.1% van de totale variantie

1.5 Bepalen van het aantal factoren

- wellicht de grootste moeilijkheid bij EFA: hoeveel factoren moeten we weerhouden?
- te weinig factoren: ‘underfactoring’; te veel factoren: ‘overfactoring’
- belangrijkste criterium: inhoudelijke interpretatie van de factoren!
- niet-inhoudelijke criteria voor het bepalen van het aantal factoren:
  - Kaiser criterium
  - Scree test
  - Parallel Analysis
  - Goodness-of-fit maten (enkel bij Maximum Likelihood extractie)
- een oneindige bron van discussie …

eigenwaarden

- een eigenwaarden-decompositie decomposeert een (vierkante) matrix in drie matrices; de middelste matrix is diagonaal en op de diagonaal staan de eigenwaarden (van groot naar klein)
- de som van de eigenwaarden is gelijk aan de som van de diagonaal elementen van de matrix; bij een correlatie-matrix is dit gelijk aan variabelen

```
> ev <- eigen(Brown2006)$values
> ev

[1] 4.2413980 1.8353059 0.4855071 0.3746198 0.3278110
[6] 0.2879887 0.2432435 0.2041260


> sum(ev)

[1] 8
```

- de eigenwaarden gedeeld door het aantal items geven het relatieve belang aan van de eerste ‘component’, de tweede component, enzoverder

```
> ev/sum(ev)
```



```
[1] 0.53017474 0.22941324 0.06068839 0.04682748 0.04097637
[6] 0.03599858 0.03040544 0.02551575
```

- ‘componenten’ zijn net als factoren, maar veronderstellen geen meetfout

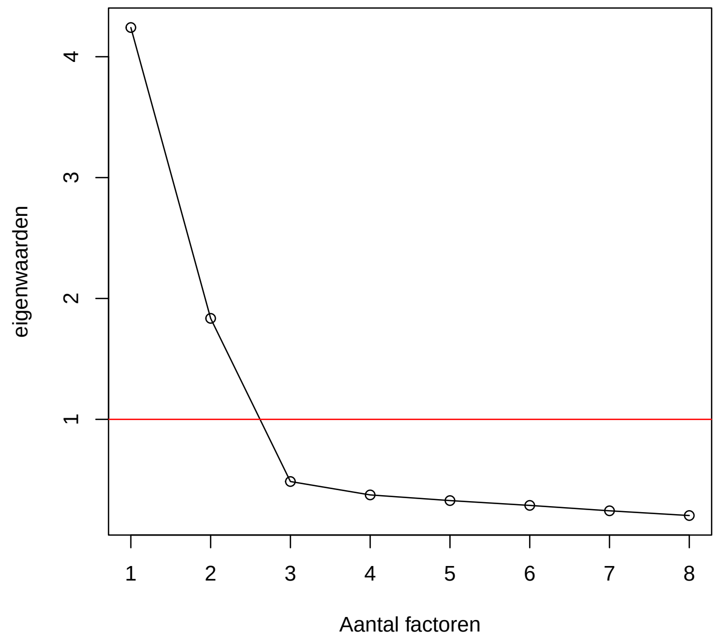
Kaiser criterium

- selecteer enkel deze factoren/componenten met een eigenwaarde groter dan 1.0
- motivatie: bij ongerelateerde indicatoren zijn de eigenwaarden (in grote steekproeven) zo goed als gelijk aan 1
- in R

```
> sum(ev > 1.0)

[1] 2
```

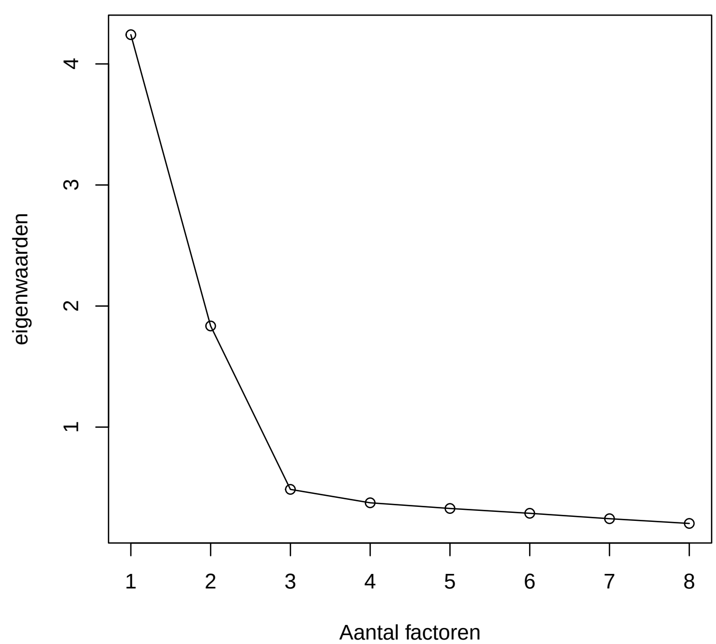
- meest gebruikt want de default in SPSS
- uit simulatiestudies blijkt dit criterium redelijk onbetrouwbaar (i.e. leidt zowel tot underfactoring als overfactoring)

Kaiser criterium: grafisch



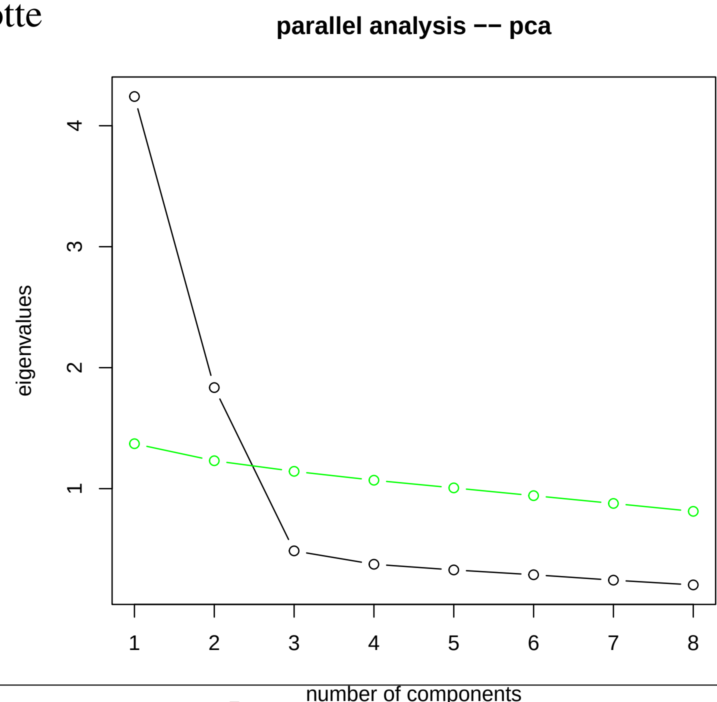
scree test

- zoek de ‘knik’ in de curve (altijd subjectief, vaak nergens te bespeuren)
- aantal factoren = aantal eigenwaarden boven de ‘knik’



‘parallel analysis’ (95 percentiel)

- rode lijn (Kaiser criterium) is vervangen door groene lijn
- groene lijn is gebaseerd op random ongecorreleerde data met zelfde steekproefgrootte



‘Exploratory graph analysis’ (EGA)

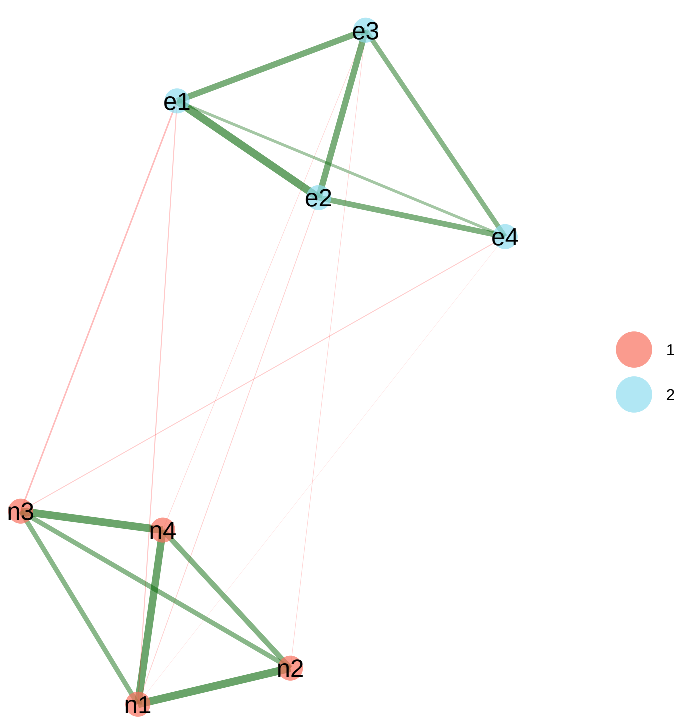
- vrij recente methode:

  Golino, H.F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS one, 12(6)*, e0174035. doi: journal.pone.0174035

- onderdeel van een nieuw veld in de psychometrie: ‘network psychometrics’
- naast het bepalen van het aantal factoren, krijgt de gebruiker ook een visualisatie van de netwerk-structuur
- in een recente simulatie (2020) blijkt deze methode het bijzonder goed te doen
- R code:

```
> library(EGAnet)
> EGA(Brown2006, n = 250)
```

```
EGA Results:

Number of Dimensions:
[1] 2

Items per Dimension:
   items dimension
n1    n1         1
n2    n2         1
n3    n3         1
n4    n4         1
e1    e1         2
e2    e2         2
e3    e3         2
e4    e4         2
```

1.6 Rotatie

- rotatie enkel vanaf meer dan 1 factor, enkel bij EFA
- rotatie is bedoeld om een (beter) interpreteerbare oplossing te bekomen
- 'simple structure' (Thurstone, 1947): een eenvoudig interpreteerbare factor-oplossing waarbij idealiter:
  1. elke factor wordt gedefinieerd door een subset van indicatoren die hoog laden op deze factor
  2. elke indicator heeft een hoge (gestandaardiseerde) lading op 1 factor ($> 0.40, > 0.50$) (*primary loading*), en zo goed als geen lading op de overige factoren (*cross-loadings*)
- rotatie heeft geen enkel effect op de communaliteiten, de unieke errorvarianties of op de evaluatie van het aantal factoren (bv. bij Maximum Likelihood blijven alle goodness-of-fit maten gelijk)
- twee soorten rotaties: *orthogonaal* en *oblique*

orthogonale rotatie

- de factoren zijn niet-gecorreleerd (i.e., de assen staan loodrecht op elkaar)
- meest gebruikt, want 'default' in SPSS! (varimax rotation)
- maar indien de latente variabelen (factoren) theoretisch gezien wel kunnen (of moeten) correleren, dan leidt een orthogonale rotatie soms tot een misleidende oplossing

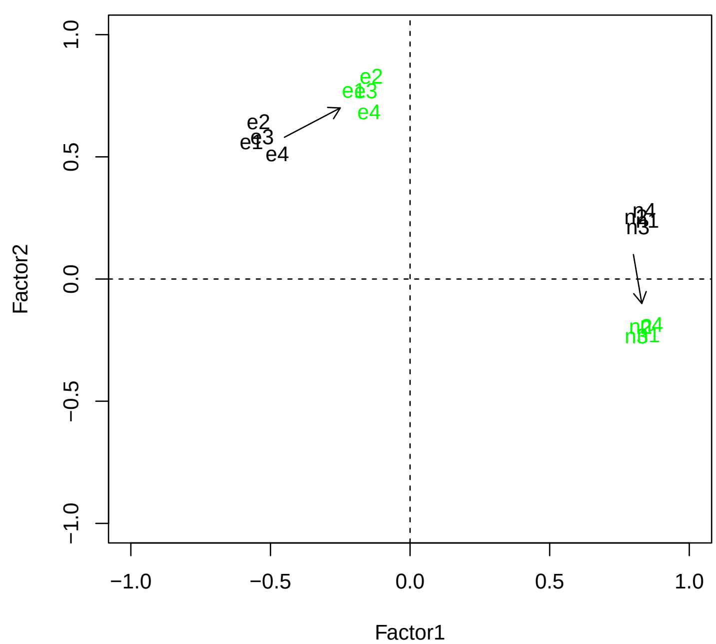
oblique rotatie

- de factoren mogen met elkaar correleren (geen loodrechte assen)
- ten onrechte weinig gebruikt!
- je krijgt ook informatie over de correlaties tussen de factoren
- verschillende methodes: promax, quartimin, oblimin, ...

Model 2b: 2 factoren, varimax rotatie

```
Call:
factanal(factors = 2, covmat = Brown2006, n.obs = N, rotation = "varimax")

Uniquenesses:
   n1    n2    n3    n4    e1    e2    e3    e4 
0.219 0.280 0.289 0.217 0.362 0.293 0.381 0.511 

Loadings:
   Factor1 Factor2
n1  0.854  -0.228 
n2  0.826  -0.194 
n3  0.811  -0.233 
n4  0.865  -0.186 
e1 -0.202   0.773 
e2 -0.139   0.829 
e3 -0.158   0.771 
e4 -0.147   0.684 

               Factor1 Factor2
SS loadings      2.923   2.526
Proportion Var   0.365   0.316
Cumulative Var   0.365   0.681

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 9.58 on 13 degrees of freedom.
The p-value is 0.728 
```


orthogonale rotatie (elke as roteert evenveel)



interpretatie

- de interpretatie van de factorladingen is nu veel duidelijker: de neuroticisme items laden vooral op de eerste factor, terwijl de extraversie items vooral laden op de tweede factor
- bemerk: de proportie verklaarde variantie van de twee factoren samen (68.1%), en de modeltoets blijven identiek
- rotatie beoogt enkel een betere interpretatie
- hier was de rotatie orthogonaal (elke as roteert evenveel)
- dit impliceert echter dat de factoren niet gecorreleerd zijn met elkaar
- meer realistisch is oblique (niet-orthogonale) rotatie

Model 2c: 2 factoren, oblique rotatie

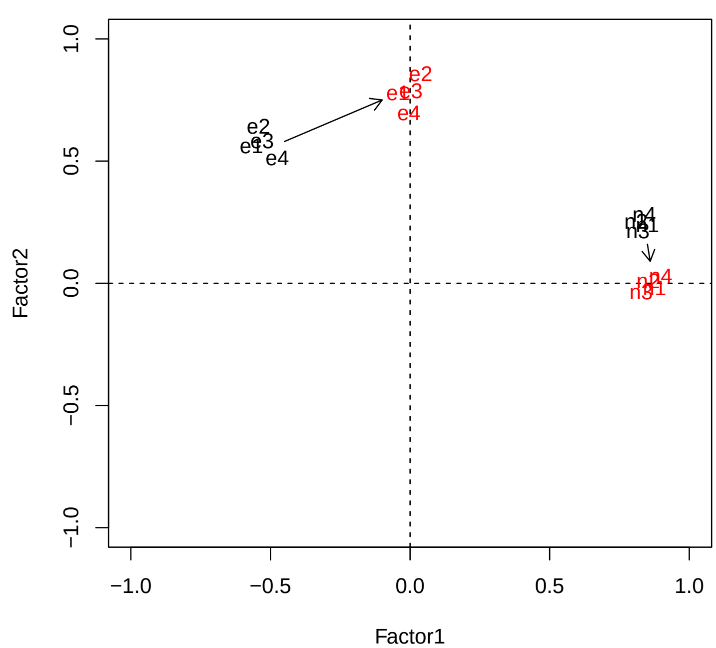
```
Call:
factanal(factors = 2, covmat = Brown2006, n.obs = N, rotation = "promax")

Uniquenesses:
   n1    n2    n3    n4    e1    e2    e3    e4
0.219 0.280 0.289 0.217 0.362 0.293 0.381 0.511

Loadings:
   Factor1 Factor2
n1  0.876
n2  0.853
n3  0.828
n4  0.898
e1          0.779
e2          0.857
e3          0.788
e4          0.698

               Factor1 Factor2
SS loadings      2.989   2.452
Proportion Var   0.374   0.307
Cumulative Var   0.374   0.680

Factor Correlations:
        Factor1 Factor2
Factor1   1.000   0.434
```

```
Factor2   0.434   1.000

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 9.58 on 13 degrees of freedom.
The p-value is 0.728
```

interpretatie

- de oblique rotatie hanteert hier het ‘promax’ algoritme
- de factorstructuur is nu nog duidelijker
- factorladingen die (in absolute waarde) kleiner zijn dan 0.1 worden weggelaten in de output, om de interpretatie te vereenvoudigen
- de modeltoets en de totale verklaarde variantie zijn (nagenoeg) identiek
- we krijgen nu ook de correlatie tussen de factoren te zien: $r = 0.434$

oblique rotatie (de ene as roteert meer dan de andere)



1.7 ($\star$)Explorative factoranalyse: via CFA

- klassieke exploratieve factoranalyse heeft enkele nadelen:
  - doorgaans worden er geen standaardfouten bij de factorladingen gerapporteerd
  - eventuele identificatieproblemen (bv. bij 'doublet factors': factoren met slechts 2 items) vallen hierdoor niet op
  - andere software voor EFA dan wel CFA staat een coherent gebruik van factoranalyse in de weg
  - CFA kan ook overweg met categorische data (bv. dichotome items)
- CFA kan ook worden gebruikt voor exploratieve factoranalyse
  - nu pas in opmars dankzij beschikbaarheid software
  - al blijft soms de vraag of een 'pure' exploratieve analyse echt wel nodig is; bij testconstructie worden items speciaal zo geselecteerd om een bepaald construct te meten; er is dus op voorhand sprake van een factorstructuur, en waarom zouden we die voorkennis niet gebruiken in de analyse door meteen CFA te hanteren?

explorative CFA in R

```
> library(lavaan)

> model <- ' efa("efa")*f1 +
+             efa("efa")*f2 =~ n1 + n2 + n3 + n4 + e1 + e2 + e3 + e4 '

> fit <- cfa(model, sample.cov = Brown2006, sample.nobs = 250,
+             rotation = "oblimin")
> summary(fit, standardized = TRUE)

lavaan 0.6-9 ended normally after 1 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        23

  Rotation method                       OBLIMIN OBLIQUE
  Oblimin gamma                                      0
  Rotation algorithm (rstarts)                GPA (100)
  Standardized metric                             TRUE
  Row weights                                     None

  Number of observations                           250

Model Test User Model:

  Test statistic                                 9.811
```

```
  Degrees of freedom                                13
  P-value (Chi-square)                           0.709

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  f1 =~ efa
    n1                0.874    0.053   16.592    0.000    0.874    0.876
    n2                0.851    0.055   15.551    0.000    0.851    0.853
    n3                0.826    0.054   15.179    0.000    0.826    0.828
    n4                0.896    0.053   16.802    0.000    0.896    0.898
    e1               -0.046    0.040   -1.138    0.255   -0.046   -0.046
    e2                0.035    0.034    1.030    0.303    0.035    0.035
    e3                0.000    0.040    0.010    0.992    0.000    0.000
    e4               -0.006    0.049   -0.131    0.896   -0.006   -0.006
  f2 =~ efa
    n1               -0.017    0.032   -0.539    0.590   -0.017   -0.017
    n2                0.011    0.035    0.322    0.748    0.011    0.011
    n3               -0.035    0.036   -0.949    0.343   -0.035   -0.035
    n4                0.031    0.031    0.994    0.320    0.031    0.031
    e1                0.776    0.059   13.125    0.000    0.776    0.778
    e2                0.854    0.058   14.677    0.000    0.854    0.855
    e3                0.785    0.060   13.106    0.000    0.785    0.787
```



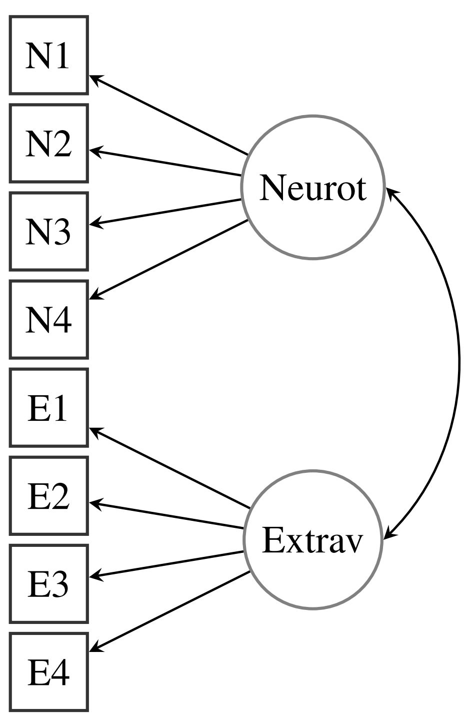
```
    e4                0.695    0.063   10.955    0.000    0.695    0.697

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  f1 ~~
    f2               -0.432    0.059   -7.345    0.000   -0.432   -0.432

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .n1                0.218    0.028    7.790    0.000    0.218    0.219
   .n2                0.279    0.032    8.693    0.000    0.279    0.280
   .n3                0.287    0.032    8.907    0.000    0.287    0.289
   .n4                0.216    0.029    7.578    0.000    0.216    0.217
   .e1                0.361    0.044    8.226    0.000    0.361    0.362
   .e2                0.292    0.043    6.787    0.000    0.292    0.293
   .e3                0.379    0.046    8.315    0.000    0.379    0.381
   .e4                0.509    0.053    9.554    0.000    0.509    0.511
    f1                1.000                               1.000    1.000
    f2                1.000                               1.000    1.000
```

1.8 ($\star$)Confirmatorische factoranalyse

- we beschikken op voorhand over een hypothese omtrent
  - het aantal factoren (en hun betekenis)
  - welke indicatoren hangen samen met elk van de factoren
  - zijn de factoren gecorreleerd of niet
- we gaan na of dit vooropgesteld model past bij de data
  - we kijken naar de modeltoets (bij voorkeur niet significant)
  - bij grotere steekproeven kijken we ook naar andere fitmaten (bv. bij voorkeur CFI > 0.95, RMSEA < 0.05, SRMR < 0.05)
- analyse vertrekt van ruwe data, of van de covariantiematrix (niet de correlatiematrix!)
- indien het model niet goed past, zijn er allerlei manieren om na te gaan waar het schoentje wringt
- technische details: zie 'toegepaste data-analyse' (1ste master)

voorbeeld 1: neuroticisme en extraversie

- het CFA model:



- omzetten correlatie matrix naar variantie-covariantie matrix (R code is geen examenstof):

```
> # correlatie matrix
> COR <- '
+  1.000
+  0.767   1.000
+  0.731   0.709   1.000
+  0.778   0.738   0.762   1.000
+ -0.351  -0.302  -0.356  -0.318   1.000
+ -0.316  -0.280  -0.300  -0.267   0.675   1.000
+ -0.296  -0.289  -0.297  -0.296   0.634   0.651   1.000
+ -0.282  -0.254  -0.292  -0.245   0.534   0.593   0.566   1.000 '
> # standaardafwijkingen voor de 8 variabelen
> sds <- c(5.7, 5.6, 6.4, 5.7, 6.0, 6.2, 5.7, 5.6)

> # aanmaken covariantie-matrix (met namen voor de variabelen)
> COV <- getCov(COR, sds = sds, names = c("n1","n2","n3","n4",
+                                          "e1","e2","e3","e4"))
> COV

          n1        n2        n3        n4       e1        e2        e3
n1  32.49000  24.48264  26.66688  25.27722 -12.0042 -11.16744  -9.61704
n2  24.48264  31.36000  25.41056  23.55696 -10.1472  -9.72160  -9.22488
n3  26.66688  25.41056  40.96000  27.79776 -13.6704 -11.90400 -10.83456
n4  25.27722  23.55696  27.79776  32.49000 -10.8756  -9.43578  -9.61704
e1 -12.00420 -10.14720 -13.67040 -10.87560  36.0000  25.11000  21.68280
```



```
e2 -11.16744  -9.72160 -11.90400  -9.43578  25.1100  38.44000  23.00634
e3  -9.61704  -9.22488 -10.83456  -9.61704  21.6828  23.00634  32.49000
e4  -9.00144  -7.96544 -10.46528  -7.82040  17.9424  20.58896  18.06672
          e4
n1  -9.00144
n2  -7.96544
n3 -10.46528
n4  -7.82040
e1  17.94240
e2  20.58896
e3  18.06672
e4  31.36000
```

- R code en output:

```
> model <- ' neuroticisme =~ n1 + n2 + n3 + n4
+            extraversie  =~ e1 + e2 + e3 + e4 '
> fit <- cfa(model, sample.cov = COV, sample.nobs = 250,
+            std.lv = TRUE, sample.cov.rescale = FALSE)
> summary(fit, standardized = TRUE, fit.measures = TRUE)

lavaan 0.6-9 ended normally after 18 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        17
```

```
  Number of observations                           250

Model Test User Model:

  Test statistic                                13.285
  Degrees of freedom                                19
  P-value (Chi-square)                           0.824

Model Test Baseline Model:

  Test statistic                              1253.791
  Degrees of freedom                                28
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    1.000
  Tucker-Lewis Index (TLI)                       1.007

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -5752.509
  Loglikelihood unrestricted model (H1)      -5745.866

  Akaike (AIC)                               11539.018
  Bayesian (BIC)                             11598.882
  Sample-size adjusted Bayesian (BIC)        11544.991
```



```
Root Mean Square Error of Approximation:

  RMSEA                                          0.000
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.034
  P-value RMSEA <= 0.05                          0.990

Standardized Root Mean Square Residual:

  SRMR                                           0.019

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  neuroticisme =~
    n1                5.043    0.289   17.472    0.000    5.043    0.885
    n2                4.752    0.291   16.337    0.000    4.752    0.849
    n3                5.399    0.333   16.190    0.000    5.399    0.844
    n4                5.027    0.289   17.381    0.000    5.027    0.882
  extraversie =~
    e1                4.811    0.333   14.465    0.000    4.811    0.802
    e2                5.169    0.338   15.294    0.000    5.169    0.834
    e3                4.500    0.318   14.150    0.000    4.500    0.789
```

```
    e4                3.915    0.327   11.974    0.000    3.915    0.699

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  neuroticisme ~~
    extraversie      -0.435    0.059   -7.410    0.000   -0.435   -0.435

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .n1                7.053    0.911    7.746    0.000    7.053    0.217
   .n2                8.782    1.003    8.755    0.000    8.782    0.280
   .n3               11.807    1.333    8.855    0.000   11.807    0.288
   .n4                7.217    0.920    7.846    0.000    7.217    0.222
   .e1               12.853    1.587    8.100    0.000   12.853    0.357
   .e2               11.718    1.609    7.285    0.000   11.718    0.305
   .e3               12.241    1.464    8.361    0.000   12.241    0.377
   .e4               16.036    1.673    9.587    0.000   16.036    0.511
    neuroticisme      1.000                               1.000    1.000
    extraversie       1.000                               1.000    1.000
```


interpretatie

- de modeltoets $\chi^2(19) = 13.285$, $p = 0.824$ is niet significant, wat wijst op een zeer goede fit
- de overige fitmaten wijzen allemaal in dezelfde richting (CFI = 1.0, RMSEA = 0.000, SRMR = 0.019)
- de kolom `Std.all` bevat de gestandaardiseerde modelparameters: eerst de factorladingen, dan de correlatie (-0.435), de unieke varianties van de indicatoren, en op het eind de varianties van de twee factoren (telkens 1.0)
- de gestandaardiseerde factorladingen zijn vrij hoog (> 0.80), met uitzondering van `e4` dat wat lager scoort dan de overige items
- de correlatie tussen de twee factoren $r = -0.435$ is negatief: hoe hoger de (latente) score op neuroticisme, hoe lager de (latente) score op extraversie; bemerk dat deze correlatie ‘vrij’ is van meetfout: er is geen nood aan een correctie voor attenuatie
- factorscores kunnen we hier niet berekenen, omdat we niet beschikken over de ruwe data

de covariantiematrix zoals voorspeld door het model

- de model-gebaseerde variantie-covariantiematrix ($\mathbf{\Sigma}$)

```
> lavInspect(fit, "Sigma")

   n1      n2      n3      n4      e1      e2      e3      e4
n1  32.490
n2  23.965  31.360
n3  27.231  25.656  40.960
n4  25.355  23.888  27.144  32.490
e1 -10.554  -9.944 -11.299 -10.520  36.000
e2 -11.340 -10.684 -12.140 -11.304  24.870  38.440
e3  -9.871  -9.300 -10.568  -9.840  21.649  23.261  32.490
e4  -8.588  -8.091  -9.193  -8.560  18.834  20.236  17.615  31.360
```

- lijkt heel erg goed op de geobserveerde variantie-covriantie matrix, vandaar de goede modelfit

(optioneel) residuen in correlatiemetriek

- een manier om problemen met het model op te sporen is om te kijken naar de verschillen tussen de geobserveerde variantie-covariantiematrix ($\mathbf{S}$) en de model-gebaseerde variantie-covariantiematrix ($\mathbf{\Sigma}$): $\mathbf{S} - \mathbf{\Sigma}$
- omdat de schaal (range) van items nogal kan verschillen, kijkt men in de praktijk vaker naar het verschil tussen de geobseerveerde en model-gebaseerde correlatiematrices:

```
> resid(fit, type = "cor")$cov

   n1     n2     n3     n4     e1     e2     e3     e4
n1  0.000
n2  0.016  0.000
n3 -0.015 -0.007  0.000
n4 -0.002 -0.010  0.018  0.000
e1 -0.042 -0.006 -0.062 -0.010  0.000
e2  0.005  0.028  0.006  0.053  0.006  0.000
e3  0.008  0.002 -0.007  0.007  0.001 -0.007  0.000
e4 -0.013  0.004 -0.035  0.023 -0.027  0.010  0.014  0.000
```

(optioneel) modeltoets

- hoe komen we aan de modeltoets?
- indien we de 'Maximum Likelihood' methode hanteren om de modelparameters te schatten wordt (voor een model zonder intercepten) de 'afstand' tussen $\boldsymbol{\Sigma}$ en $\mathbf{S}$ berekend volgens deze formule:

$$F_{ML} = \log|\boldsymbol{\Sigma}| + \text{tr}(\mathbf{S}\boldsymbol{\Sigma}^{-1}) - \log|\mathbf{S}| - p$$

- n keer deze waarde voor F_{ML} geeft de waarde van de modeltoets:

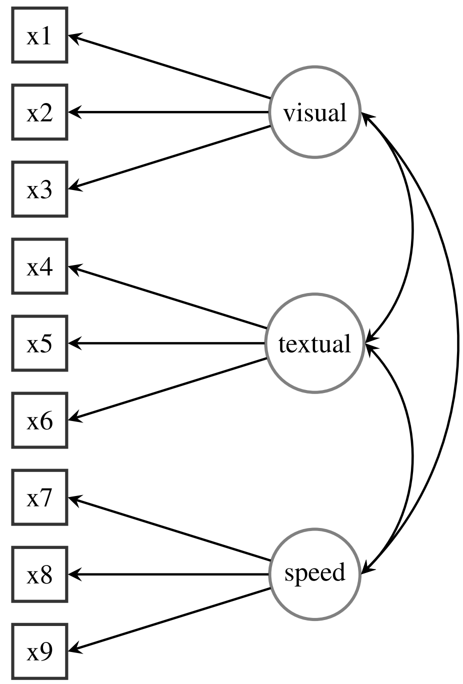
```
> Sigma <- lavInspect(fit, "Sigma")
> S <- lavInspect(fit, "samplestats")$cov
> p <- nrow(S)
> F.ML <- log(det(Sigma)) + sum(diag(S %*% solve(Sigma))) - log(det(S)) - p
> X2 <- nobs(fit) * F.ML
> X2

[1] 13.28499
```


voorbeeld 2: de Holzinger en Swineford data

- 'klassieke' dataset; gebaseerd op data van Holzinger & Swineford (1939)
- ook geanalyzeerd door Jöreskog (1969)
- 9 geobserveerde indicatoren (subtesten) die verondersteld worden om 3 latente variabelen te meten:
  - een 'visual' factor, gemeten door x1, x2 en x3
  - een 'textual' factor, gemeten door x4, x5 en x6
  - een 'speed' factor gemeten door x7, x8 en x9
- N=301
- we veronderstellen dat de drie factoren onderling gecorreleerd zijn
- voor meer info omtrent deze dataset, tik in R:

```
> ?HolzingerSwineford1939
```

grafische voorstelling 3-factor model



cfa in R

```
> HS.model <- ' visual  =~ x1 + x2 + x3
+               textual =~ x4 + x5 + x6
+               speed   =~ x7 + x8 + x9 '
> fit <- cfa(HS.model, data=HolzingerSwineford1939,
+            std.lv = TRUE)
> summary(fit, fit.measures = TRUE, standardized = TRUE)

lavaan 0.6-9 ended normally after 20 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        21

  Number of observations                           301

Model Test User Model:

  Test statistic                                85.306
  Degrees of freedom                                24
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                               918.852
  Degrees of freedom                                36
  P-value                                        0.000
```

```
User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.931
  Tucker-Lewis Index (TLI)                       0.896

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -3737.745
  Loglikelihood unrestricted model (H1)      -3695.092

  Akaike (AIC)                                7517.490
  Bayesian (BIC)                              7595.339
  Sample-size adjusted Bayesian (BIC)         7528.739

Root Mean Square Error of Approximation:

  RMSEA                                          0.092
  90 Percent confidence interval - lower         0.071
  90 Percent confidence interval - upper         0.114
  P-value RMSEA <= 0.05                          0.001

Standardized Root Mean Square Residual:

  SRMR                                           0.065

Parameter Estimates:
```



```
  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                0.900    0.081   11.128    0.000    0.900    0.772
    x2                0.498    0.077    6.429    0.000    0.498    0.424
    x3                0.656    0.074    8.817    0.000    0.656    0.581
  textual =~
    x4                0.990    0.057   17.474    0.000    0.990    0.852
    x5                1.102    0.063   17.576    0.000    1.102    0.855
    x6                0.917    0.054   17.082    0.000    0.917    0.838
  speed =~
    x7                0.619    0.070    8.903    0.000    0.619    0.570
    x8                0.731    0.066   11.090    0.000    0.731    0.723
    x9                0.670    0.065   10.305    0.000    0.670    0.665

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual ~~
    textual           0.459    0.064    7.189    0.000    0.459    0.459
    speed             0.471    0.073    6.461    0.000    0.471    0.471
  textual ~~
    speed             0.283    0.069    4.117    0.000    0.283    0.283

Variances:
```

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
.x1	0.549	0.114	4.833	0.000	0.549	0.404
.x2	1.134	0.102	11.146	0.000	1.134	0.821
.x3	0.844	0.091	9.317	0.000	0.844	0.662
.x4	0.371	0.048	7.779	0.000	0.371	0.275
.x5	0.446	0.058	7.642	0.000	0.446	0.269
.x6	0.356	0.043	8.277	0.000	0.356	0.298
.x7	0.799	0.081	9.823	0.000	0.799	0.676
.x8	0.488	0.074	6.573	0.000	0.488	0.477
.x9	0.566	0.071	8.003	0.000	0.566	0.558
visual	1.000				1.000	1.000
textual	1.000				1.000	1.000
speed	1.000				1.000	1.000

interpretatie

- modeltoets $\chi^2(24) = 85.306$, $p \approx 0$ wijst op een slechte fit; de CFI = 0.931 is wat meer optimistisch (> 0.90 wijst op een redelijke fit), maar RMSEA = 0.092 en SRMR = 0.065 (deze laatste twee hebben we graag kleiner dan 0.05, of toch zeker kleiner dan 0.06)
- ondanks het ‘klassieke’ statuut van deze dataset, past een 3-factor model niet erg goed bij de data
- dit zou een goede aanleiding zijn om opnieuw ‘exploratief’ de data te bestuderen
  - (optioneel) binnen het CFA kader: modification indices en residuen laten toe mogelijk oorzaken van misfit te ontdekken
  - exploratief: we beginnen gewoon opnieuw met exploratieve factoranalyse
- verdere interpretatie van de modelparameters heeft niet erg veel zin: het model past niet eens goed bij de data!

factorscores CFA model

- enkel ter illustratie tonen we hoe je factorscores kunt berekenen

```
> head(lavPredict(fit), n = 15)

           visual     textual       speed
 [1,] -0.90891107 -0.13897739  0.09928895
 [2,]  0.05504432 -1.02327052  1.00971612
 [3,] -0.84635334 -1.89178446 -1.35691369
 [4,]  0.46613216  0.01867815 -0.43801293
 [5,] -0.46231182 -0.12352320  0.31370005
 [6,]  0.02585069 -1.34366599  1.14428189
 [7,] -1.09487107  0.66446221  0.07055461
 [8,]  0.11876593 -0.10622272 -0.85928082
 [9,] -0.28230949  0.40854178  0.32344799
[10,] -1.17184874  0.09388268 -0.91475568
[11,] -0.84311428 -0.77740732 -0.52097364
[12,]  0.44250506  0.09721354 -0.58681806
[13,]  0.57688997 -0.14184920  0.46344428
[14,]  0.26798578  0.21799490 -0.39161729
[15,]  0.78096739  1.17003359 -0.20943148
```


1.9 (⋆)Factoranalyse van de ‘Satisfaction With Life’ schaal

- inlezen data

```
> # inlezen data via bestand op eigen pc
> # DataSWL <- read.csv("c:/temp/DataSWL.csv")

> # inlezen data via internet
> DataSWL <- read.csv("https://www.da.ugent.be/datasets/DataSWL.csv")
```

- exploratieve factoranalyse, 1 factor

```
> fit1 <- factanal(DataSWL, factors = 1)
> fit1

Call:
factanal(x = DataSWL, factors = 1)

Uniquenesses:
 sat1  sat2  sat3  sat4  sat5
0.326 0.606 0.195 0.881 0.486

Loadings:
     Factor1
sat1 0.821
sat2 0.628
sat3 0.897
```

```
sat4 0.345
sat5 0.717

               Factor1
SS loadings      2.507
Proportion Var   0.501

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 118.81 on 5 degrees of freedom.
The p-value is 5.61e-24
```

- modeltoets niet erg rooskleurig, probeer 2 factoren met oblique rotatie

```
> fit2 <- factanal(DataSWL, factors = 2, rotation = "promax")
> fit2

Call:
factanal(x = DataSWL, factors = 2, rotation = "promax")

Uniquenesses:
 sat1  sat2  sat3  sat4  sat5
0.169 0.568 0.249 0.005 0.351

Loadings:
     Factor1 Factor2
sat1  0.975  -0.200
sat2  0.701  -0.137
```



```
sat3  0.802   0.136
sat4 -0.151   1.050
sat5  0.527   0.431

               Factor1 Factor2
SS loadings      2.386   1.365
Proportion Var   0.477   0.273
Cumulative Var   0.477   0.750

Factor Correlations:
        Factor1 Factor2
Factor1   1.000   0.408
Factor2   0.408   1.000

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 8.35 on 1 degree of freedom.
The p-value is 0.00386
```

- hm, beter maar `sat5` laadt zowel op de eerste als de tweede factor? ook de modeltoets is nog steeds niet goed
- misschien drie factoren? (niet mogelijk met slechts 5 indicatoren)

```
> fit3 <- try(factanal(DataSWL, factors = 3))
```

- 2 factoren met exploratieve CFA

```
> model <- ' efa("efa")*f1 +
+             efa("efa")*f2 =~ sat1 + sat2 + sat3 + sat4 + sat5 '
> fit <- cfa(model, DataSWL, rotation = "oblimin")
> summary(fit)

lavaan 0.6-9 ended normally after 26 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        14

  Rotation method                      OBLIMIN OBLIQUE
  Oblimin gamma                                      0
  Rotation algorithm (rstarts)               GPA (100)
  Standardized metric                             TRUE
  Row weights                                     None

  Number of observations                           200

Model Test User Model:

  Test statistic                                 4.252
  Degrees of freedom                                 1
  P-value (Chi-square)                           0.039

Parameter Estimates:

  Standard errors                             Standard
```



```
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 =~ efa
    sat1              1.437    0.103   13.998    0.000
    sat2              1.075    0.107   10.018    0.000
    sat3              1.055    0.077   13.772    0.000
    sat4             -0.001    0.005   -0.223    0.823
    sat5              0.964    0.099    9.747    0.000
  f2 =~ efa
    sat1             -0.192    0.062   -3.116    0.002
    sat2             -0.167    0.075   -2.226    0.026
    sat3              0.107    0.053    2.034    0.042
    sat4              2.205    0.544    4.053    0.000
    sat5              0.427    0.142    3.011    0.003

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 ~~
    f2                0.197    0.067    2.939    0.003

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .sat1              0.596    0.130    4.594    0.000
   .sat2              1.447    0.160    9.073    0.000
   .sat3              0.369    0.064    5.749    0.000
```

```
    .sat4             -2.379    2.412   -0.986    0.324
    .sat5              1.000    0.149    6.692    0.000
     f1                1.000
     f2                1.000
```

- negatieve errorvariantie (voor sat4) en grote standaardfout (2.412) suggereren een (identificatie)probleem wat verdoken bleef bij de `factanal` functie
- maar inhoudelijk kunnen we twee factoren vooropstellen: de eerste drie items richten zich meer op 'present satisfaction', terwijl de laatste twee items eerder peilen naar 'satisfaction with the past'
- we voeren een confirmatorische factoranalyse uit:

```
> model <- ' present =~ sat1 + sat2 + sat3
+            past    =~ sat4 + sat5 '
> fit <- cfa(model, data = DataSWL, std.lv = TRUE)
> summary(fit, standardized = TRUE)

lavaan 0.6-9 ended normally after 28 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        11
```



```
  Number of observations                           200

Model Test User Model:

  Test statistic                                49.714
  Degrees of freedom                                 4
  P-value (Chi-square)                           0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  present =~
    sat1              1.420    0.095   14.883    0.000    1.420    0.883
    sat2              1.061    0.105   10.056    0.000    1.061    0.663
    sat3              1.043    0.075   13.872    0.000    1.043    0.841
  past =~
    sat4              0.654    0.137    4.790    0.000    0.654    0.415
    sat5              2.078    0.273    7.616    0.000    2.078    1.378

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  present ~~
```

```
    past              0.495    0.087    5.666    0.000    0.495    0.495

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .sat1              0.573    0.114    5.033    0.000    0.573    0.221
   .sat2              1.435    0.160    8.952    0.000    1.435    0.561
   .sat3              0.452    0.071    6.409    0.000    0.452    0.294
   .sat4              2.052    0.234    8.779    0.000    2.052    0.827
   .sat5             -2.043    1.148   -1.780    0.075   -2.043   -0.898
    present           1.000                               1.000    1.000
    past              1.000                               1.000    1.000
```

- fit blijft ondermaats, en ook hier zien we een negatieve variantie terug (dit keer voor `sat5`); op basis van de gestandaardiseerde factorladingen blijkt het item `sat4` niet goed te passen bij de 2-de factor
- optioneel: 'modification indices' geven suggesties om de modelfit te verbeteren; de drie belangrijkste voor dit model zijn:

```
> modindices(fit, sort = TRUE, max = 3)

    lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
26 sat3 ~~ sat4 37.569  0.470   0.470    0.488    0.488
21 sat1 ~~ sat4 17.070 -0.416  -0.416   -0.384   -0.384
24 sat2 ~~ sat4  7.166 -0.302  -0.302   -0.176   -0.176
```


- dit zijn telkens correlaties tussen de errortermen van enkele items; bemerk dat item `sat4` telkens terug opduikt
- besluit: een grondige herziening van item 'sat4' (en mogelijks de hele schaal) dringt zich op!